Fuzzy-rough cognitive networks

Peer-reviewed author version

NAPOLES RUIZ, Gonzalo; Mosquera, Carlos; Falcon, Rafael; Grau, Isel; Bello, Rafael & VANHOOF, Koen (2018) Fuzzy-rough cognitive networks. In: NEURAL NETWORKS, 97, p. 19-27.

DOI: 10.1016/j.neunet.2017.08.007
Handle: http://hdl.handle.net/1942/25527

Fuzzy-Rough Cognitive Networks

Gonzalo Nápoles[a], Carlos Mosquera[b], Rafael Falcon[c], Isel Grau[b], Rafael Bello[b], Koen Vanhoof[a]

[a]*Faculty of Business Economics, Universiteit Hasselt, Belgium*
[b]*Department of Computer Sciences, Central University of Las Villas, Cuba*
[c]*School of Electrical Engineering and Computer Science, University of Ottawa, Canada*

Abstract

Rough Cognitive Networks (RCNs) are a kind of granular neural network that augment the reasoning rule present in Fuzzy Cognitive Maps with crisp information granules coming from Rough Set Theory. While RCNs have shown promise in solving different classification problems, this model is still very sensitive to the similarity threshold upon which the rough information granules are built. In this paper, we cast the RCN model within the framework of *fuzzy rough sets* in an attempt to eliminate the need for a user-specified similarity threshold while retaining the model's discriminatory power. As far as we know, this is the first study that brings fuzzy sets into the domain of rough cognitive mapping. Numerical results in presence of 140 well-known pattern classification problems reveal that our approach, referred to as *Fuzzy-Rough Cognitive Networks*, is capable of outperforming most traditional classifiers used for benchmarking purposes. Furthermore, we explore the impact of using different heterogeneous distance functions and fuzzy operators over the performance of our granular neural network.

Key words: fuzzy cognitive maps, fuzzy rough sets, rough cognitive mapping, pattern classification, granular classifiers

Email addresses: `gonzalo.napoles@uhasselt.be` (Gonzalo Nápoles), `xcarlos@gmail.com` (Carlos Mosquera), `rfalc032@uottawa.ca` (Rafael Falcon), `grau.isel@gmail.com` (Isel Grau), `rbellop@uclv.edu.cu` (Rafael Bello), `koen.vanhoof@uhasselt.be` (Koen Vanhoof)

1. Introduction

Pattern classification [1] is one of the most popular field within Artificial Intelligence as a result of its link with real-world problems. In short, it may be defined as the process of identifying the right category (among those in a predefined set) to which an observation belongs. The ease with which we recognize our beloved black cat from hundreds similar to it or read hand-written characters belies the astoundingly complex processes that underlie these scenarios. That is why researchers have been focused on developing a wide spectrum of classification algorithms called *classifiers* with the goal of solving these problems with the best possible accuracy.

The literature on classification models [2] is vast and offers a myriad of techniques that approach the classification problem from multiple angles. Regrettably, some of the most accurate classifiers do not provide any mechanism to explain how they arrived at each conclusion and behave like *black boxes*. This means that their reasoning mechanism is not transparent, therefore negatively affecting their practical usability in scenarios where understanding the decision process is required. According to the terminology discussed in [3], *transparency* can be understood as the classifier's ability to explain its reasoning mechanism, whereas *interpretability* refers to the classifier's ability to explain the problem domain at the attribute level.

Recently, Nápoles and his collaborators [4] introduced the *Rough Cognitive Networks* (RCNs) in an attempt to develop an accurate, transparent classifier. Such granular neural networks augment the reasoning scheme present in Fuzzy Cognitive Maps (FCMs) [5] with information granules coming from Rough Set Theory (RST) [6]. Although RCNs can be considered as recurrent neural systems that fit the McCulloch-Pitts' scheme [7], there are important differences with regards to other neural models.

Classical neural networks regularly perform like black boxes, where neither neurons nor connections have any clear specific meaning for the problem itself [8]. However, all the neurons and connections in an RCN have a precise meaning at a granular level, therefore making it possible to understand the underlying decision process at a granular (symbolic) level. The absence of hidden neurons and the lazy learning approach are other distinctive features attached to these granular, recurrent neural systems.

While RCNs have shown promise in solving different pattern classification problems [4] [9], their performance is still very sensitive to an input parameter denoting the similarity threshold upon which the rough information granules are built. The proper estimation of this parameter is essential in presence of numerical attributes since it defines whether two objects are deemed similar or not. Aiming at overcoming this drawback, Nápoles et. al. [4] proposed an optimization-based hyperparameter learning scheme to estimate the value of this parameter from a hold-out test set. However, this strategy may become impractical for large datasets since it requires rebuilding the information granules for each parameter value to be evaluated.

In [10] the authors proposed a granular ensemble named *Rough Cognitive Ensembles* (RCEs) to deal with the parametric requirements of RCN-based classifiers. This classification model employs a collection of RCNs, each operating at a different granularity degree. While this approach involves a more elaborated solution, the ensemble architecture and the bagging strategy used to improved the diversity among the base classifiers irremediably harm the transparency of RCNs, thus becoming another black-box.

In this paper, we cast the RCN approach within the framework of Fuzzy Rough Set Theory (FRST) [11] [12] [13] [14] in an attempt to eliminate the need for a user-specified similarity threshold while retaining the model's discriminatory power. Fuzzy rough sets are an extension of classical rough sets in which fuzzy sets are used to characterize the degree to which an object belongs to each information granule. The inclusion of the fuzzy approach into the RCN model allows coping with both the vagueness (fuzzy sets) and inconsistency (rough sets) of the information typically found in pattern classification environments. Besides, it allows designing a more elegant solution for the parametric issues of RCN-based classifiers.

Numerical simulations using 140 datasets reveal that the proposed model, referred to as *Fuzzy-Rough Cognitive Networks* (FRCNs), is capable of outperforming the standard RCNs using a fixed, reasonable similarity threshold value. The results also suggest that FRCNs remain competitive with regards to RCEs and other black-box classifiers adopted for comparison purposes. More importantly, the challenging process of estimating a precise value for the similarity threshold parameter is no longer a concern.

The rest of this paper is organized as follows. Section 2 briefly describes the RCN algorithm and the motivation behind our proposal. The fuzzy RCN classifier is unveiled in Section 3, whereas Section 4 introduces the numerical simulations and their ensuing discussion. Towards the end, Section 5 outlines some concluding remarks and future work directions.

2. Rough Cognitive Mapping

This section discusses the technical background relevant to this study and explains the motivation behind the fuzzy approach.

2.1. Theoretical Background

Rough cognitive mapping is a recently introduced concept[4] that brings together RST and FCMs. RCNs are granular FCMs whose topology is defined by the abstract semantics of the three-way decision rules [15] [16]. The set of input neurons in an RCN represent the positive, boundary and negative regions of the decision classes in the problem under consideration. The output neurons describe the set of decision classes. The topology (both concepts and weights) is entirely computed from historical data, thus removing the need for expert intervention during the classifier's construction.

The first step in the RCN learning process is related to the *input data granulation* using RST. The positive, boundary and negative regions of each decision class according to a subset of attributes are computed using the training data set and a predefined similarity relation.

The second step is concerned with *topology design* where a sigmoid FCM is automatically created from the discovered information granules by using a set of predefined rules; see [4] for more details. In principle, an RCN will be composed of at most $4|\mathcal{D}|$ neurons and $3|\mathcal{D}|(1+|D|)$ causal relationships, with $\mathcal{D} = \{D_1, \ldots, D_K\}$ being the set of decision classes.

The last step refers to the *network exploitation*, which simply means computing the response vector $\mathcal{A}_x(\mathcal{D}) = \{A_x(D_1), \ldots, A_x(D_k), \ldots, A_x(D_K)\}$ for some unlabeled object. The new object x is presented to the RCN as an input vector $A^{(0)}$ that activates input neurons. Each element in $A^{(0)}$ is computed on the basis of the inclusion degree of x to each rough granular region. After this, the input vector is propagated through the RCN using the McCulloch-Pitts reasoning model [7] and next the decision class with the highest value in the response vector is then assigned to the test object.

2.2. Motivation for the FRCN Approach

The notion of *rough cognitive mapping* opened up a new research avenue in the field of granular-neural classifiers. However, their performance is highly sensitive to the similarity threshold used to determine whether two instances can be gathered together into the same similarity class.

Nápoles et. al. [4] used a parameter tuning method based on the Harmony Search (HS) optimizer to estimate the similarity threshold. Nevertheless, the evaluation of every candidate solution requires recalculating the lower and upper approximations of each RST-based region for each decision class, which could be computationally prohibitive for large datasets.

Let us assume that $\mathcal{U}_1 \subset \mathcal{U}$ is the training set and $\mathcal{U}_2 \subset \mathcal{U}$ is the hold-out test (validation) set such that $\mathcal{U}_1 \cap \mathcal{U}_2 = \emptyset$. The computational complexity of building the lower and upper approximations is $O(|\Phi||\mathcal{U}_1|^2)$, with Φ being the attribute set, whereas the complexity of building the network topology is $O(|\mathcal{D}|^2)$, with $\mathcal{D}$ being the set of decision classes. Besides, the complexity of exploiting the granular network for $|\mathcal{U}_2|$ instances is $O(|\mathcal{U}_2||\Phi||\mathcal{U}_1|^2)$. This implies that the temporal complexity of evaluating a single parameter value is $O(\max\{|\Phi||\mathcal{U}_1|^2, |\mathcal{D}|^2, |\mathcal{U}_2||\Phi||\mathcal{U}_1|^2\})$. Due to the fact that $|\mathcal{U}_1| \geq |\mathcal{U}_2|$ in most machine learning scenarios, we can conclude that the overall complexity of this parameter learning method is $O(T|\Phi||\mathcal{U}_1|^3)$, where T is the number of learning cycles. Regrettably, this may negatively affect the practical usability of RCNs in solving real-world pattern classification problems.

The key goal behind this research is to remove the estimation of the similarity threshold without affecting the overall RCN's discriminatory power. Being more explicit, we aim to arrive at a parameterless classifier (and hence suppressing the need for a parameter tuning strategy) without degrading the RCN's performance in classification problems.

3. Fuzzy-Rough Cognitive Mapping

This section presents the notion of *fuzzy-rough cognitive mapping* in order to remove the requirement of estimating the similarity threshold in an RCN. With this goal in mind, we first describe the mathematical foundations behind this approach. Afterwards, we explain how to construct an FRCN for solving pattern classification problems.

3.1. Fuzzy-Rough Set Theory

The hybridization between rough sets and fuzzy sets was originally investigated by Dubois and Prade [11], and later extended and/or modified by several authors. In this paper, we adopt the approach proposed by Inuiguchi et al. [14] since it includes some mathematical properties that may be convenient when designing our fuzzy-rough classifier.

Let us assume the universe $\mathcal{U}$, a fuzzy set $X \in \mathcal{U}$ and a fuzzy binary relation $P \in \mathcal{F}(\mathcal{U} \times \mathcal{U})$, where $\mu_X(x)$ and $\mu_P(y,x)$ denote their respective membership functions. The function $\mu_X : \mathcal{U} \rightarrow [0,1]$ computes the membership degree to which $x \in \mathcal{U}$ is a member of X, whereas $\mu_P : \mathcal{U} \times \mathcal{U} \rightarrow [0,1]$ denotes the degree to which y is presumed to be a member of X from the fact that x is a member of the fuzzy set X. For the sake of simplicity, $P(x)$ is defined by its membership function $\mu_{P(x)}(y) = \mu_P(y,x)$.

In order to define the lower and upper approximations of a set in fuzzy environments, we should consider the consistency degree of x being a member of X under the knowledge P. This degree can be measured by the truth value of the statement "$y \in P(x)$ implies $y \in X$" under fuzzy sets $P(x)$ and X. To do that, we use a necessity measure $\inf_{y\in\mathcal{U}} \mathcal{I}(\mu_P(y,x), \mu_{X_k}(y))$ with an implication function $\mathcal{I} : [0,1] \times [0,1] \rightarrow [0,1]$ such that $\mathcal{I}(0,0) = \mathcal{I}(0,1) = \mathcal{I}(1,0) = \mathcal{I}(1,1) = 0$, where $\mathcal{I}(.,a)$ decreases and $\mathcal{I}(a,.)$ increases, $\forall a \in [0,1]$. In this formulation, X_k is the set comprising all objects labeled with the k-th decision class. Equation (1) displays the membership function for the lower approximation $P_*(X)$ associated with the fuzzy set X.

$$\mu_{P_*(X_k)}(x) = \min\left\{\mu_{X_k}(x), \inf_{y\in\mathcal{U}} \mathcal{I}(\mu_P(y,x), \mu_{X_k}(y))\right\} \tag{1}$$

Analogously to the lower approximation, we can derive the membership function for the upper approximation assuming that X is a fuzzy set and P is a fuzzy binary relation. By doing so, we should measure the truth value of the statement "$\exists y \in \mathcal{U}$ such that $x \in P(y)$" under fuzzy sets $P(x)$ and X. The true value of this statement can be obtained by a possibility measure $\sup_{y\in\mathcal{U}} \mathcal{T}(\mu_P(x,y), \mu_{X_k}(y))$ with a conjunction function $\mathcal{T} : [0,1] \times [0,1] \rightarrow [0,1]$ such that $\mathcal{T}(0,0) = \mathcal{T}(0,1) = \mathcal{T}(1,0) = \mathcal{T}(1,1) = 0$, where both $\mathcal{T}(.,a)$ and $\mathcal{T}(a,.)$ increase, $\forall a \in [0,1]$. Equation (2) shows the membership function for the upper approximation $P^*(X)$ associated with X.

$$\mu_{P^*(X_k)}(x) = \max\left\{\mu_{X_k}(x), \sup_{y\in\mathcal{U}} \mathcal{T}(\mu_P(x,y), \mu_{X_k}(y))\right\} \tag{2}$$

It should be remarked that the intersection of two fuzzy sets X and Y is regularly defined as $\mu_{X\cap Y} = \min\{\mu_X(x), \mu_Y(x)\}, \forall x \in \mathcal{U}$, whereas their union takes the form $\mu_{X\cup Y} = \max\{\mu_X(x), \mu_Y(x)\}, \forall x \in \mathcal{U}$. However, some researchers replace the *min* operator with a t-norm and the *max* operator with a t-conorm [14]. On the other hand, note that Inuiguchi's model does not assume that $\mu_P(x,x)=1, \forall x \in \mathcal{U}$. Instead, we compute the minimum between $\mu_X(x)$ and $\inf_{y\in\mathcal{U}} \mathcal{I}(\mu_P(y,x), \mu_{X_k}(y))$ when computing $\mu_{P_*(X_k)}(x)$, and the maximum between $\mu_X(x)$ and $\sup_{y\in\mathcal{U}} \mathcal{T}(\mu_P(x,y), \mu_{X_k}(y))$ when computing $\mu_{P^*(X_k)}(x)$. This feature allows preserving the inclusiveness of $P_*(X)$ in the fuzzy set X and the inclusiveness of X in $P^*(X)$.

Based on the above elements, one can define the three fuzzy-rough regions that comprise the core of the granulation stage. Equation (3), (4) and (5) display the membership functions associated with the fuzzy-rough positive, negative and boundary regions, respectively.

$$\mu_{POS(X_k)}(x) = \mu_{P_*(X_k)}(x) \tag{3}$$

$$\mu_{NEG(X_k)}(x) = 1 - \mu_{P_*(X_k)}(x) \tag{4}$$

$$\mu_{BND(X_k)}(x) = \mu_{P^*(X_k)}(x) - \mu_{P_*(X_k)}(x) \tag{5}$$

These memberships functions allow computing more flexible information granules. As such, abrupt transitions between classes are replaced with gradual ones, therefore allowing an element to belong to more than one class with varying degrees. Next, we explain how to exploit these fuzzy-rough information granules by using a cognitive neural network.

3.2. Fuzzy-Rough Cognitive Networks

The proposed FRCN model transforms the attribute space into a fuzzy-rough one, which is exploited by a recurrent neural network. Under these fuzzy conditions, objects are categorized into information granules with soft boundaries, and therefore, a strict similarity threshold is no longer required. This suggests that the first step when constructing an FRCN is related with the fuzzy granulation of the available information.

Let $X = \{X_1, \ldots, X_k, \ldots, X_M\}$ be a partition of $\mathcal{U}$ according to the values of the decision attribute such that the subset X_k comprises those objects labeled as D_k. Based on this partition, we can define the membership degree of $x \in \mathcal{U}$ to a subset X_k (see Equation 6). We assume that all objects labeled as D_k have maximum membership degree to the k-th subset; however, more sophisticated variants can be formalized as well.

$$\mu_{X_k}(x) = \begin{cases} 1 & , y \in X_k \\ 0 & , y \notin X_k \end{cases} \tag{6}$$

Another pivotal element to be defined is the membership function $\mu_P(y, x)$ associated with the fuzzy binary relation. Equation (7) shows the function adopted in this paper, which depends on the membership degree of object x to X, and the similarity degree between x and y. The similarity degree $\varphi(x, y)$ denotes the complement of the normalized distance $\delta(x, y)$ between two instances x and y. Section 4.2 describes some heterogeneous distance functions explored in this study that allow comparing instances comprising both numerical and nominal attributes.

$$\mu_P(y, x) = \mu_{X_k}(x)\varphi(x, y) = \mu_{X_k}(x)(1 - \delta(x, y)) \tag{7}$$

Let us assume that the universe of discourse $\mathcal{U}$ is composed of those objects comprised into the training dataset and $\Theta : \mathcal{U} \rightarrow \mathcal{D}$ is a function that returns the decision class attached to each training instance, such that $\mathcal{D} = \{D_1, \ldots, D_K\}$. Algorithm 1 summarizes the steps for granulating the information space under the fuzzy settings described above.

Algorithm 1. Fuzzy-rough information granulation.

```
FOREACH x ∈ U DO
    IF Θ(x) = D_k THEN
        X_k ← X_k ∪ {x}
    END IF
    Compute μ_{X_k}(x) according to Equation 6
END
FOREACH x ∈ U DO
    FOREACH subset X_k DO
        Compute μ_{POS(X_k)}(x) according to Equation 3
        Compute μ_{NEG(X_k)}(x) according to Equation 4
        Compute μ_{BND(X_k)}(x) according to Equation 5
    END
END
```

After granulating the information space, the resultant fuzzy-rough constructs are used to build a neural network. Similarly to RCN models, input neurons denote positive or negative fuzzy-rough regions, whereas output neurons comprise the decision classes for the problem at hand. During preliminary simulations we noticed that including the fuzzy-rough boundary regions into the modeling did not significantly increase the classifier's discriminatory ability. This behavior is not surprising because in crisp-rough environments the hesitant evidence is more conclusive when compared to the evidence coming from fuzzy-rough granules. Therefore, the neural network topology can be constructed by using the following rules:

- (R_1^*) IF $C_i = P_k^*$ AND $C_j = D_k$ THEN $w_{ij} = 1.0$
- (R_2^*) IF $C_i = N_k^*$ AND $C_j = D_k$ THEN $w_{ij} = -1.0$
- (R_2^*) IF $C_i = P_k^*$ AND $C_j = D_{v \neq k}$ THEN $w_{ij} = -1.0$
- (R_4^*) IF $C_i = P_k^*$ AND $C_j = P_{v \neq k}$ THEN $w_{ij} = -1.0$

where C_i is the i-th neural processing entity, D_k represents k-th decision class, while P_k^* and N_k^* are neurons denoting the positive and negative fuzzy-rough region associated to the k-th decision class.

Figure 1 shows the network topology of FRCNs for binary classification problems. Without loss of generality, any FRCN comprises $2|\mathcal{D}|$ input neurons, $|\mathcal{D}|$ output neurons and $|\mathcal{D}|(4 + |\mathcal{D}|)$ causal weights. Observe that the number of neurons in the causal network is not determined by the number of features but by the number of decision classes.

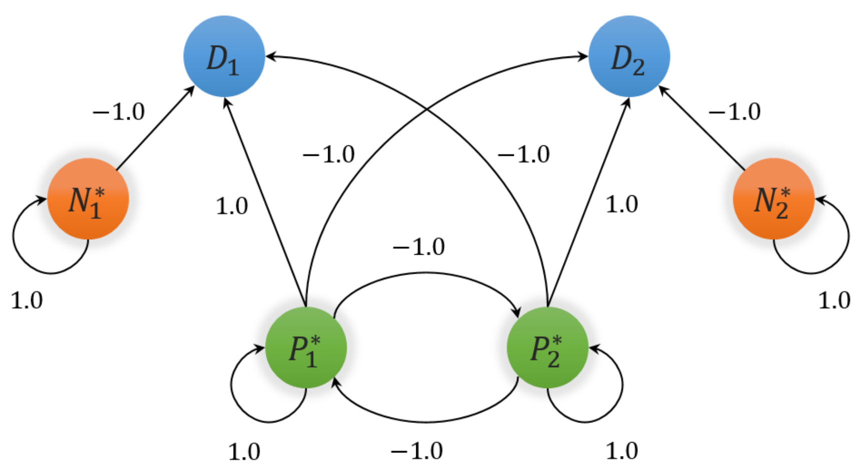

Figure 1: Fuzzy-Rough Cognitive Network for binary classification problems.

Algorithm 2 shows the steps required to build the topology of the granular neural network from discovered information granules.

Algorithm 2. Network construction procedure.

```
FOREACH subset X_k DO
   Add a neuron P_k as the kth positive region
   Add a neuron N_k as the kth positive region
   Add a neuron B_k as the kth positive region
END
FOREACH decision D_k DO
   Add a neuron D_k as the kth decision
END
FOREACH neuron C_i DO
   FOREACH neuron C_j DO
      Configure w_ij according to rules R*_1 - R*_4
   END
END
```

Once the network has been constructed, we can perform the classification for new (unlabeled) instances by activating the input-type neurons and performing the reasoning process. In order to activate these neurons, we use the similarity degree between the object y and $x \in \mathcal{U}$ as well as the membership degree of x to each fuzzy-rough granular region.

Figure 2 and 3 illustrate the semantics behind this activation mechanism for the k-th positive and negative region, respectively. More explicitly, such figures show the degree to which y belongs to the fuzzy intersection defined from the membership functions $\mu_{POS(X_k)}(x)$ (or $\mu_{NEG(X_k)}(x)$), and the fuzzy similarity relation between the new instance y and $x \in X$. As a further step, we calculate the inclusion degree of the fuzzy intersection set into the k-th fuzzy-rough region. This procedure produces a normalized value that will be used to activate the input neurons in the causal network.

Equation (8) formalizes a generalized measure to compute the activation value of the k-th positive neuron, where $\mathcal{T}_2$ denotes a t-norm, $\varphi(x, y)$ is the similarity degree between x and y whereas $\mu_{POS(X_k)}(x)$ represents the membership grade of x to the k-th fuzzy-rough positive region. A t-norm is a conjunction function $\mathcal{T}_2 : [0,1] \times [0,1] \to [0,1]$ that fulfills three conditions: (i) $\forall a \in [0,1], \mathcal{T}_2(a,1) = \mathcal{T}_2(1,a) = a$, (ii) $\forall a,b \in [0,1], \mathcal{T}_2(a,b) = \mathcal{T}_2(b,a)$, and (iii) $\forall a,b,c \in [0,1], \mathcal{T}_2(a, \mathcal{T}_2(b,c)) = \mathcal{T}_2(\mathcal{T}_2(a,b),c)$. Similarly, we can activate neurons denoting fuzzy-rough negative regions. Only output neurons remain inactive at the outset of the neural reasoning process.

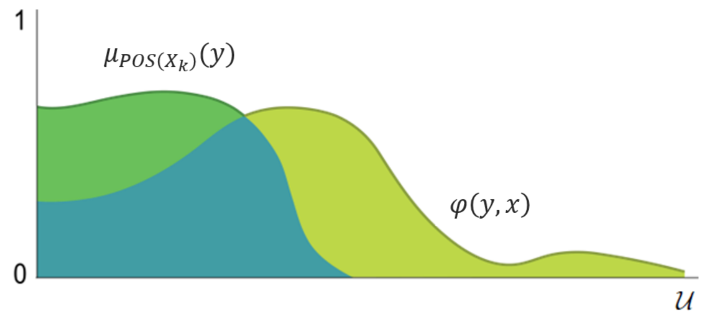

Figure 2: Inclusion degree of y into the k-th positive region.

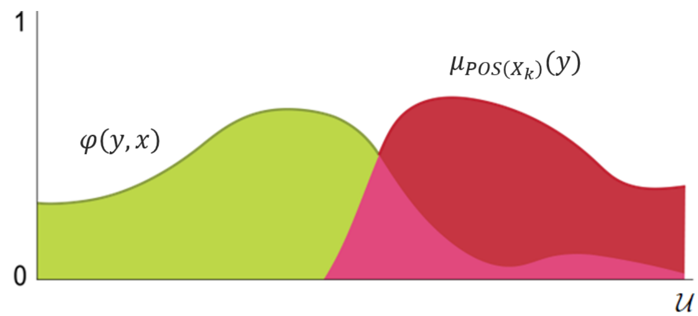

Figure 3: Inclusion degree of y into the k-th negative region.

$$\mathcal{A}(P_k^*) = \frac{\int \mathcal{T}_2(\varphi(x,y), \mu_{POS(X_k)}(x))dx}{\int \mu_{POS(X_k)}(x)dx} \tag{8}$$

However, due to the fact that the universe of discourse $\mathcal{U}$ is rather finite, the use of integrals may not be convenient. Rules (R_5^*) and (R_6^*) show a more practical mechanism to activate the granular classifier.

- (R_5^*) IF $C_i = P_k^*$ THEN $A_i^{(0)} = \frac{\sum_{x\in\mathcal{U}} \mathcal{T}_2(\varphi(x,y), \mu_{POS(X_k)}(x))}{\sum_{x\in\mathcal{U}} \mu_{POS(X_k)}(x)}$

- (R_6^*) IF $C_i = N_k^*$ THEN $A_i^{(0)} = \frac{\sum_{x\in\mathcal{U}} \mathcal{T}_2(\varphi(x,y), \mu_{NEG(X_k)}(x))}{\sum_{x\in\mathcal{U}} \mu_{NEG(X_k)}(x)}$

Once the initial activation vector $A^{(0)}$ associated with the object y has been computed, we perform the neural reasoning process until (i) a fixed-point attractor is discovered, or alternatively (ii) a maximal number of iterations is reached. At that point, the label of the output neuron having the highest activation value is assigned to the target object.

Algorithm 3a shows the first step towards exploiting the neural network, that is, the activation of input neurons for a new test instance x. Similarly, Algorithm 3b illustrates how to determine the decision class from outputs neurons once the input neurons have been activated.

Algorithm 3a. Network activation procedure.

FOREACH decision D_k DO
    Calculate $A_x^{(0)}(P_k)$ according to rule R_5^*
    Calculate $A_x^{(0)}(N_k)$ according to rule R_6^*
END

Algorithm 3b. Network reasoning procedure.

FOR $t = 0$ TO T DO
    $converged \leftarrow TRUE$
    FOREACH neuron C_i DO
        Compute $A_i^{(t+1)} = f\left(\sum_{j=1}^{M} w_{ji} A_j^{(t)}\right)$
        IF $A_i^{(t)} \neq A_i^{(t+1)}$ THEN
            $converged \leftarrow FALSE$
        END
    END
    IF $converged$ THEN
        RETURN $argmax_k\{\mathcal{A}_x^{(t+1)}(D_k)\}$
    END
END
IF not $converged$ THEN
    RETURN $argmax_k\{\mathcal{A}_x^{(T)}(D_k)\}$
END

It is worth mentioning that the FRCN algorithm can operate in either a lazy or inductive fashion. In a lazy setting, both the fuzzy-rough granules and the network topology can be constructed when the new instance arrives. This is however not efficient since the granules and the topology can be reused to classify new instances. In the inductive approach, the knowledge is stored into the discovered granules and the causal weight matrix, which is prescriptively determined by construction rules (R_1^*) - (R_4^*). Adjusting such causal weights using a supervised learning algorithm is a promising research direction to be explored as a future work.

4. Numerical Simulations

In this section, we conduct several simulations to evaluate the predictive capability of the proposed fuzzy-rough neural network. As a first experiment, we investigate the impact of using different fuzzy operators and distance functions across 140 pattern classification data sets. Afterward, we compare the prediction capability of the best-performing fuzzy-rough model against 17 well-established state-of-the-art classifiers.

4.1. Dataset Characterization

We leaned upon 140 classification datasets taken from the KEEL [17] and UCI ML [18] repositories. These problems comprise different characteristics and allow evaluating the predictive power of both state-of-the-art and the granular classifiers under consideration.

In the adopted datasets [1], the number of attributes ranges from 2 to 262, the number of decision classes from 2 to 100, and the number of instances from 14 to 12,906. They involve 13 noisy and 47 imbalanced datasets, with the imbalance ratio fluctuating between 5:1 and 2160:1. To avoid the out-of-range issues, the numerical attributes have been normalized. Furthermore, we replaced missing values with the mean or the mode depending on whether the attribute was numerical or nominal, respectively.

As a final element, each dataset has been partitioned using a standard 10-fold cross-validation procedure, i.e., each problem has been split into 10 folds, each containing 10% of the instances.

4.2. Heterogeneous Distance Functions

The distance function plays a pivotal role when designing instance-based classifiers. Next, we briefly describe three distance functions [19] [10] used in our experiments that allow comparing heterogeneous instances, i.e., objects comprising both numerical and nominal attributes.

- *Heterogeneous Euclidean-Overlap Metric (HEOM).* This distance function computes the normalized Euclidean distance between numerical attributes and an overlap metric for nominal attributes.

[1] The reader can find a complete characterization of such datasets in [10]

- *Heterogeneous Manhattan-Overlap Metric (HMOM).* This heterogeneous variant is similar to the HEOM function but it replaces the Euclidean distance with the Manhattan distance when computing the dissimilarity between two numerical values.

- *Heterogeneous Value Difference Metric (HVDM).* This function involves a stronger strategy for quantifying the dissimilarity between nominal attributes. Instead of using a matching approach, it measures the correlation between attributes and the decision classes.

4.3. Determining the Best-Performing Fuzzy Model

The first experiment is oriented to determining the combination of fuzzy operators leading to the best prediction rates. The FRCN algorithm requires the specification of a fuzzy implicator and two t-norms. The $\mathcal{I}$ implicator is used to compute the membership degree of an object to the lower approximations, the $\mathcal{T}_1$ t-norm is used to compute the membership degree of an object to the upper approximations whereas the $\mathcal{T}_2$ t-norm is used to activate the neural processing entities. For the sake of simplicity, we use the same t-norm to compute the membership degree to the upper approximations as well as to exploit the neural network. Tables 1 and 2 display the t-norms and fuzzy implicators included in this first simulation.

Table 1: T-norms explored in this paper.

T-norm	**Formulation**
Standard intersection	$\mathcal{T}(x,y)=\min\{x,y\}$
Algebraic product	$\mathcal{T}(x,y)=xy$
Lukasiewicz	$\mathcal{T}(x,y)=\max\{0,x+y-1\}$
Drastic product	$\mathcal{T}(x,y)=\begin{cases} x & ,y=1 \\ y & ,x=1 \\ 0 & ,otherwise \end{cases}$

To measure the classifiers' prediction capability, we computed the Kappa coefficient. Cohen's Kappa coefficient [20] measures the inter-rater agreement for categorical items. It is usually deemed a more robust measure than the standard accuracy since this coefficient takes into account the agreement occurring by chance. Figure 4 shows the average Kappa coefficient achieved by each model for different combinations of fuzzy operators using the HMOM distance as the standard dissimilarity functional.

Table 2: Fuzzy implicators explored in this paper.

Implicator	**Formulation**
Standard	$\mathcal{I}(x,y) = \begin{cases} 1 & , x \leq y \\ 0 & , x > y \end{cases}$
Kleene-Dienes	$\mathcal{I}(x,y) = \max\{1-x, y\}$
Lukasiewicz	$\mathcal{I}(x,y) = \min\{1-x+y, 1\}$
Zadeh	$\mathcal{I}(x,y) = \max\{1-x, \min\{x,y\}\}$
Godel	$\mathcal{I}(x,y) = \begin{cases} 1 & , x \leq y \\ y & , x > y \end{cases}$
Larsen	$\mathcal{I}(x,y) = xy$
Mamdani	$\mathcal{I}(x,y) = \min\{x,y\}$
Reichenbach	$\mathcal{I}(x,y) = 1-x+xy$
Yager	$\mathcal{I}(x,y) = \begin{cases} 1 & , x = y = 0 \\ y^x & , otherwise \end{cases}$
Goguen	$\mathcal{I}(x,y) = \begin{cases} 1 & , x \leq y \\ y/x & , otherwise \end{cases}$

From the above results we can notice that the FRCN method computes the best prediction rates when the Lukasiewicz t-norm is used to activate the input neurons regardless of the fuzzy operator attached to the membership functions $\mu_{P_*(X_k)}(x)$ and $\mu_{P^*(X_k)}(x)$. Consequently, we adopt the Lukasiewicz implicator and the Lukasiewicz t-norm as standard fuzzy operators in the rest of the simulations conducted in this paper.

The following experiment is devoted to comparing the prediction capability of the proposed classifier with respect to the crisp variant (RCN) using a reasonably, fixed similarity threshold equal to 0.98. Figure 5 summarizes the average Kappa measure attained by each classifier for different distance functions. The simulations confirm that the FRCN models always report better prediction rates regardless of the underlying distance function, although the HMOM function seems to stand as the best choice.

Aiming at conducting a more rigorous analysis, we compute the Friedman two-way analysis of variances by ranks [21]. The test advocates for the rejection of the null hypothesis (p-value $= 8.1268E-10 < 0.05$) for a confidence interval of 95%, hence we can conclude that there are significant differences between at least two models across datasets.

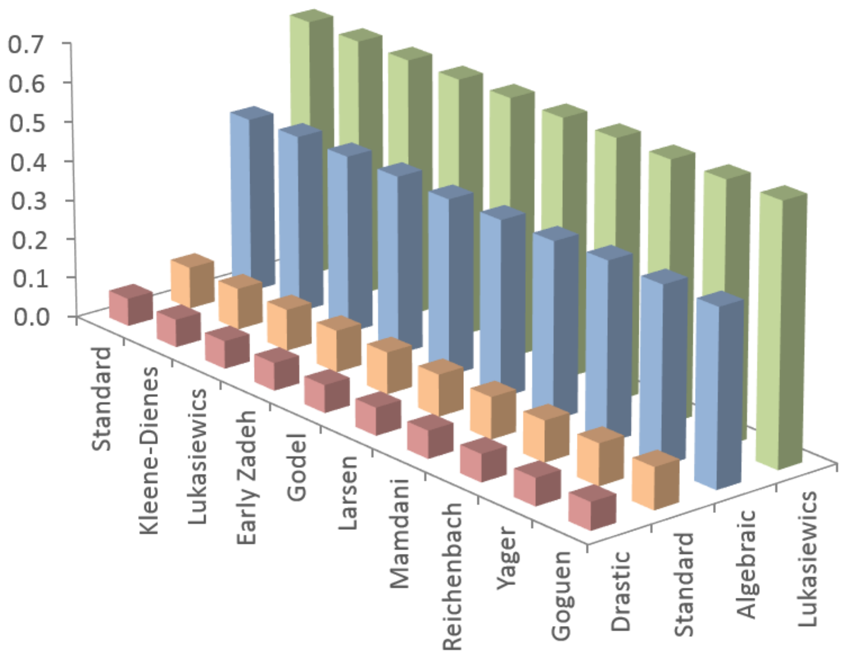

Figure 4: Average Kappa measure computed for the proposed fuzzy-rough classifier using the HMOM distance function with different t-norm and fuzzy implicators.

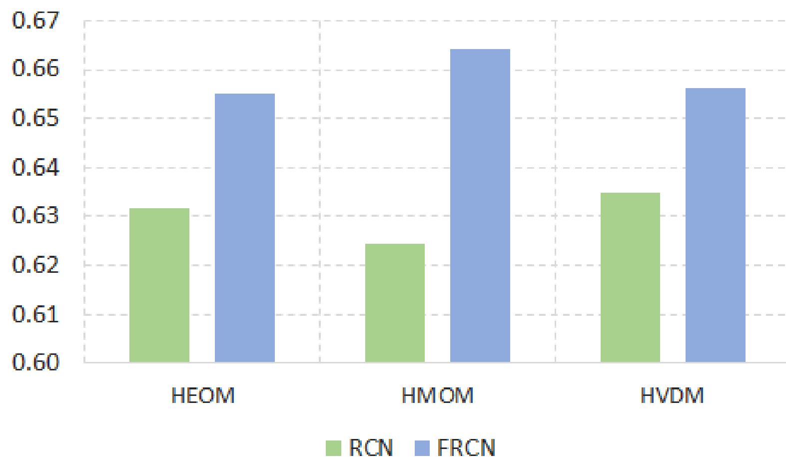

Figure 5: Average Kappa measure for different distance functions.

The next experiment is focused on determining whether the superiority of the FRCN-HMOM classifier is statistically significant or not. To that end, we resorted to the Wilcoxon signed rank test [22] and post-hoc procedures to adjust the p-values instead of using mean-ranks approaches, as recently suggested by Benavoli and collaborators [23].

Table 3 reports the unadjusted p-value computed by the Wilcoxon signed rank test and the corrected p-values associated with each pairwise comparison using FRCN-HMOM as the control method. In this paper, we assume that a null hypothesis can be rejected if at least one of the post-hoc procedures advocates for the rejection. The statistical analysis confirms FRCN-HMOM's superiority as all the null hypotheses were rejected.

Table 3: Adjusted p-values according to different post-hoc procedures using the best-performing rough classifier (FRCN-HMOM) as the control method.

Algorithm	p-value	Bonferroni	Holm	Holland	Null Hypothesis
RCN-HEOM	2.15E-07	0.000001	0.000001	0.000001	Rejected
RCN-HMOM	2.50E-07	0.000001	0.000001	0.000001	Rejected
RCN-HVDM	0.000003	0.000015	0.000009	0.000009	Rejected
FRCN-HEOM	0.000076	0.000380	0.000152	0.000152	Rejected
FRCN-HVDM	0.007897	0.039485	0.007897	0.007897	Rejected

The above simulations suggest that the proposed FRCN algorithm, like the *Rough Cognitive Ensembles* [10], is capable of suppressing the parametric requirements of RCNs without harming their performance. But are they similar in performance? In order to answer this question we can use the Wilcoxon signed rank test for pairwise comparisons. The test suggests accepting the conservative hypothesis (p-value=0.7387 $>$ 0.05) using a confidence interval of 95%. Therefore, we can conclude that both approaches perform similarly for the datasets adopted in the empirical comparison.

However, the fuzzy approach proposed in this paper is preferred since it fits best the parsimony principle: *the simpler the better*. The bagging scheme and the ensemble model itself make the RCE algorithm less transparent than the fuzzy variant, thus notably reducing one of the main contributions attached to rough cognitive classifiers. In other words, we can achieve the same prediction rates by using a single fuzzy-rough classifier rather than an ensemble composed of several crisp models!

4.4. Comparison Against State-of-the-Art Classifiers

In this section, we compare the prediction ability of the best-performing fuzzy model (FRCN-HMOM, hereinafter simply called FRCN) against the following well-known state-of-the-art classifiers:

- **Rule-based models**
  - *Decision Table (DT)* [24]. The algorithm searches for matches in the body using a subset of attributes. If no instances are found, the majority class in the table is returned; otherwise, the majority class of all matching instances is returned.
- **Bayesian models**
  - *Naïve Bayes (NB)* [25]. A probabilistic classification algorithm using estimator classes, where numeric estimator precision values are chosen based on the analysis of the training data.
  - *Naïve Bayes Updateable (NBU)* [25]. Implements an incremental NB classifier that learns one instance at a time. Instead of using normal density measures for numerical attributes, this algorithm employs a kernel estimator without discretization.
- **Function-based models**
  - *Simple Logistic (SL)* [26]. A classifier building linear logistic regression models. LogitBoost with simple regression functions as base learners is used for fitting the logistic models.
  - *Multilayer Perceptron (MLP)* [27]. Neural network that uses the backpropagation algorithm to train the model.
  - *Support Vector Machines (SMO)*. [28] Implements John Platt's sequential minimal optimization algorithm for training a support vector classifier. In our research, we adopted a quadratic polynomial kernel to perform the numerical simulations.
- **Tree-based models**
  - *Decision Tree (J48)* [29]. Induces classification rules in the form of a pruned/unpruned decision tree.

- *Random Tree (RT)* [30]. Decision tree without pruning that considers k randomly chosen attributes at each node.
- *Random Forest (RF)* [31]. Bagging of random trees.
- *Fast Decision Tree (FDT)* [32]. Builds a tree using information gain and prunes it using reduced-error pruning.
- *Best-first Decision Tree (BFT)* [33]. Classification trees that use binary split for both nominal and numeric attributes.
- *Logistic Model Tree (LMT)* [34]. Decision trees for classification that use logistic regression functions at the leaves.

- **Instance-based models**
  - *Nearest Neighbor (NN)* [35]. Instance-based (lazy) classifier that simply chooses the closest instance to the test instance and returns its class.
  - *k-Nearest Neighbors (kNN)* [35]. Lazy learner that computes the predicted class based upon the classes of the k training instances that are most similar to the test instance, as determined by a similarity function.
  - *K^* classifier (K^*)* [36]. Instance-based classifier similar to kNN that uses an entropy-based distance function.

- **Fuzzy-rough models**
  - *Fuzzy-Rough k-Nearest Neighbors (FRNN)* [37]. Nearest neighbor model that utilizes the lower and upper approximations from fuzzy rough set theory to classify test instances.
  - *Vaguely-quantified k-Nearest Neighbors (VQNN)* [38]. Fuzzy-rough model that emulates the linguistic quantifiers *some* and *most* when performing the classification process.

In our simulations, we retain the default parameter settings implemented in Weka v3.6.11 [39], therefore no classification algorithm explicitly performs parameter tuning. Despite the fact that a proper parametric setting often increases the algorithm's performance over multiple data sources [40], a robust classifier should be able to produce good results even when its parameters might not have been optimized for a specific problem.

Analogously to the previous simulations, we utilized Cohen's Kappa coefficient to quantify the algorithms' performance. Figure 6 displays the average Kappa measure attained by each classification algorithm across the selected datasets. The results show that FRCN is the second-best ranked algorithm whereas LMT arises as the best-performing classifier.

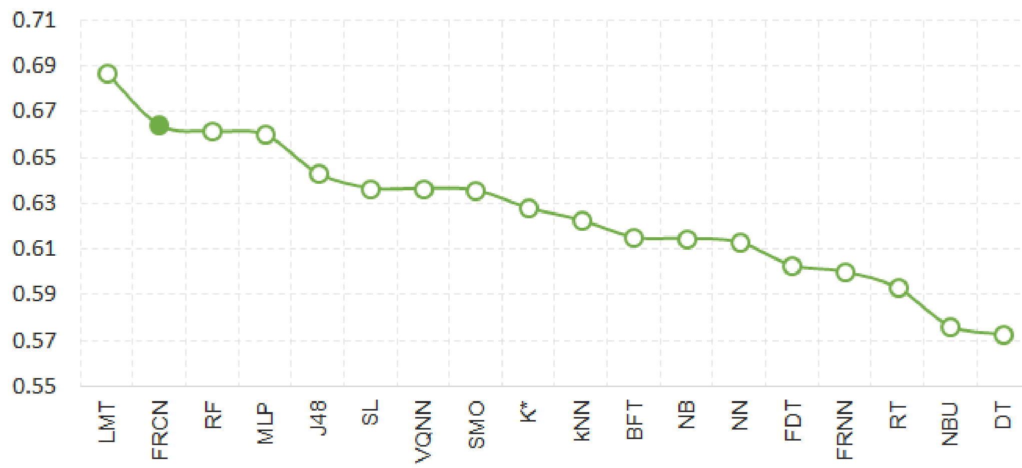

Figure 6: Average Kappa values reported by each classifier.

For this experiment, the Friedman test suggests rejecting the null hypothesis (p-value $= 1.4396928E-10 < 0.1$) for a confidence interval of 90%. This suggests that there are significant differences between at least two algorithms across the 140 datasets adopted for simulation.

Table 4 summarizes the p-values reported by the Wilcoxon signed rank test and the corrected p-values according to several post-hoc procedures using FRCNs as the control method. The results indicate that LMT is the best-performing classifier in our study, with no significant differences spotted between our proposal and MLP, RF, SMO and SL, as the null hypothesis was accepted in each of these pairwise comparisons.

The superiority of LMT is quite interesting. This method allows inducing trees with linear-logistic regression models at the leaves. During the training process, it determines the appropriate number of boosting iterations by internally cross-validating the model until the performance ceases to increase. This is somehow similar to the RCNs' parameter tuning step that our fuzzy approach attempts suppressing, so one may question whether including the LMT algorithm in our simulations is fair at all.

Table 4: Adjusted p-values according to different post-hoc procedures using the proposed fuzzy-rough classifier (FRCNs) as the control method.

Algorithm	p-value	Bonferroni	Holm	Holland	Null Hypothesis
RT	1.29E-11	2.34E-10	2.21E-10	2.21E-10	Rejected
DT	4.94E-11	8.91E-10	7.92E-10	7.91E-10	Rejected
NBU	3.79E-08	6.82E-07	5.68E-07	5.68E-07	Rejected
FDT	6.31E-07	1.13E-05	0.000008	8.83E-06	Rejected
FRNN	8.94E-07	0.000016	0.000011	1.16E-05	Rejected
NN	2.15E-06	0.000038	0.000025	2.57E-05	Rejected
NB	5.51E-06	0.000099	0.000060	6.06E-05	Rejected
BFT	3.20E-05	0.000576	0.000320	0.000319	Rejected
kNN	7.13E-05	0.001285	0.000642	0.000642	Rejected
K^*	0.005752	0.103543	0.046019	0.045103	Rejected
LMT	0.006376	0.114778	0.046019	0.045103	Rejected
J48	0.010528	0.189511	0.063170	0.061531	Rejected
VQNN	0.010947	0.197052	0.063170	0.061531	Rejected
SL	0.109578	1.000000	0.438314	0.371388	Failed to reject
SMO	0.273587	1.000000	0.820761	0.616689	Failed to reject
RF	0.940694	1.000000	1.000000	0.996482	Failed to reject
MLP	1.000000	1.000000	1.000000	1.000000	Failed to reject

On the other hand, FRCN's superiority upon other instance-based classifiers such as kNN or K* is remarkable. We conjecture that this could be a direct result of using all the available evidence to infer the most likely decision for a new instance, instead of only using the information contributed by the positive region (e.g., the k closest neighbors). Combining such evidence in a nonlinear manner as the FRCN neurons do is likely another key piece towards the attainment of high prediction rates.

Equally important is the fact that our classification algorithm provides an introspection mechanism into its decision process, which stands as its chief advantage over comparably accurate black-box classifiers. It is fair to mention that the literature includes several neural models that provide such explanatory features. For example, the *Evolving Fuzzy Neural Networks* [41], the *Dynamic Evolving Neural-Fuzzy Inference System* [42] and the *Evolving Spiking Neural Networks* [43] all rely on low-level fuzzy rules to extract knowledge from the problem domain. This cannot be naturally achieved with our high-level approach. However, in presence of high-dimensional problems, these algorithms induce a large number of fuzzy rules with many antecedents, which are difficult to interpret in practice. The number of causal rules codified into an FRCN does not depend on the number of attributes but on the number of decision classes in the problem at hand. This guarantees that the introspection mechanism attached to FRCNs remains fairly interpretable and unaffected by the problem dimensionality.

5. Conclusions

In this paper, we introduced the notion of *fuzzy-rough cognitive mapping* in an attempt to get rid of the parameter learning requirements of RCN-based models. In the FRCN algorithm, information granules have soft boundaries, thus leading to gradual transitions between the classes as opposed to abrupt transitions that regularly occur in crisp environments.

The results have shown that the proposed fuzzy classifier is capable of outperforming the crisp RCN variant regardless of the adopted distance function. In spite of that, the Lukasiewicz operators and the HMOM distance function stand as the best choices. From the comparison between the best-performing fuzzy model and 17 state-of-the-art classifiers, we concluded that FRCNs are as accurate as the most successful black boxes. The main advantage of our granular neural network relies on its ability to elucidate its decision process using inclusion degrees and causal relations. It is worth mentioning that our classifier performs better than other instance-based learners across the datasets adopted for simulation purposes.

More importantly, the results support the hypothesis behind our research: that the fuzzy-rough approach allows completely suppressing the parametric requirements behind rough cognitive mapping without either harming its performance or significantly increasing its computational complexity.

Of course, the classifier presented in this paper is *no panacea*. While the foundations underpinning FRCNs seem quite intuitive for mathematicians, it may not be intuitive enough for experts with no background in Computer Science or related areas. Besides, computing a transparent decision model does not necessarily imply that we can understand the problem domain at a low level. As a future work, we will investigate other strategies to automatically construct FCM-based classifiers from historical data. Deriving FCM-based models with lower abstraction levels leads to truly interpretable classifiers although their accuracy may be compromised.

Acknowledgments

This work was supported by the Research Council of Hasselt University. The authors would like to thank the anonymous reviewers for their constructive remarks throughout the revision process.

References

[1] R. O. Duda, P. E. Hart, D. G. Stork, Pattern classification, 2nd Edition, John Wiley & Sons, 2012.

[2] I. H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Second Edition, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[3] G. Nápoles, Rough Cognitive Networks, Ph.D. thesis, Hasselt University (May 2017).

[4] G. Nápoles, I. Grau, E. Papageorgiou, R. Bello, K. Vanhoof, Rough Cognitive Networks, Knowledge-Based Systems 91 (2016) 46–61.

[5] B. Kosko, Fuzzy cognitive maps, International Journal Man-Machine Studies 24 (1) (1986) 65–75.

[6] Z. Pawlak, Rough sets, International Journal of Computer & Information Sciences 11 (5) (1982) 341–356.

[7] W. S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, in: J. A. Anderson, E. Rosenfeld (Eds.), Neurocomputing: Foundations of Research, MIT Press, 1988, pp. 15–27.

[8] G. Nápoles, E. Papageorgiou, R. Bello, K. Vanhoof, On the convergence of sigmoid Fuzzy Cognitive Maps, Information Sciences 349 (2016) 154–171.

[9] G. Nápoles, I. Grau, R. Falcon, R. Bello, K. Vanhoof, A Granular Intrusion Detection System using Rough Cognitive Networks, in: R. Abielmona, R. Falcon, N. Zincir-Heywood, H. Abbass (Eds.), Recent Advances in Computational Intelligence in Defense and Security, Springer Verlag, 2016, Ch. 7, pp. 169–191.

[10] G. Nápoles, R. Falcon, E. Papageorgiou, R. Bello, K. Vanhoof, Rough cognitive ensembles, International Journal of Approximate Reasoning 85 (2017) 79–96.

[11] D. Dubois, H. Prade, Rough fuzzy sets and fuzzy rough sets, International Journal of General Systems 17 (1990) 91–209.

[12] A. M. Radzikowska, E. E. Kerre, A comparative study of fuzzy rough sets, Fuzzy Sets and Systems 126 (2) (2002) 137–155.

[13] C. Cornelis, M. De Cock, A. M. Radzikowska, Fuzzy rough sets: from theory into practice, Handbook of Granular Computing (2008) 533–552.

[14] M. Inuiguchi, W.-Z. Wu, C. Cornelis, N. Verbiest, Fuzzy-Rough Hybridization, Springer Berlin Heidelberg, 2015, pp. 425–451.

[15] Y. Yao, Three-way decision: an interpretation of rules in rough set theory, in: P. Wen, Y. Li, L. Polkowski, Y. Yao, S. Tsumoto, G. Wang (Eds.), Rough Sets and Knowledge Technology, Springer Verlag, 2009, pp. 642–649.

[16] Y. Yao, The superiority of three-way decisions in probabilistic rough set models, Information Sciences 181 (6) (2011) 1080–1096.

[17] J. Alcalá, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework, Journal of Multiple-Valued Logic and Soft Computing 17 (2-3) (2011) 255–287.

[18] M. Lichman, UCI machine learning repository (2013).
URL `http://archive.ics.uci.edu/ml`

[19] D. R. Wilson, T. R. Martinez, Improved heterogeneous distance functions, Journal of Artificial Intelligence Research 6 (1) (1997) 1–34.

[20] B. D. Eugenio, M. Glass, The kappa statistic: a second look, Computational Linguistics 30 (1) (2004) 95–101.

[21] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, Journal of the american statistical association 32 (200) (1937) 675–701.

[22] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics 1 (1945) 80–83.

[23] A. Benavoli, G. Corani, F. Mangili, Should we really use post-hoc tests based on mean-ranks?, Journal of Machine Learning Research 17 (2016) 1–10.

[24] R. Kohavi, The power of decision tables, in: Machine Learning: ECML-95, Springer, 1995, pp. 174–189.

[25] G. H. John, P. Langley, Estimating continuous distributions in bayesian classifiers, in: Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc., 1995, pp. 338–345.

[26] M. Sumner, E. Frank, M. Hall, Speeding up logistic model tree induction, in: Knowledge Discovery in Databases: PKDD 2005, Springer, 2005, pp. 675–683.

[27] R. Hecht-Nielsen, Theory of the backpropagation neural network, in: International Joint Conference on Neural Networks, IEEE, 1989, pp. 593–605.

[28] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, K. R. K. Murthy, Improvements to platt's smo algorithm for svm classifier design, Neural Computation 13 (3) (2001) 637–649.

[29] J. R. Quinlan, C4.5: programs for machine learning, Morgan Kauffman Publishers, 1993.

[30] Y. Amit, D. Geman, Shape quantization and recognition with randomized trees, Neural Computation 9 (7) (1997) 1545–1588.

[31] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.

[32] J. Su, H. Zhang, A fast decision tree learning algorithm, in: Proceedings of the 21st National Conference on Artificial Intelligence, AAAI'06, AAAI Press, 2006, pp. 500–505.

[33] H. Shi, Best-first decision tree learning, Ph.D. thesis, Citeseer (2007).

[34] N. Landwehr, M. Hall, E. Frank, Logistic model trees, Machine Learning 59 (1-2) (2005) 161–205.

[35] D. W. Aha, D. Kibler, M. K. Albert, Instance-based learning algorithms, Machine learning 6 (1) (1991) 37–66.

[36] J. G. Cleary, L. E. Trigg, et al., K*: An instance-based learner using an entropic distance measure, in: Proceedings of the 12th International Conference on Machine learning, Vol. 5, 1995, pp. 108–114.

[37] R. Jensen, C. Cornelis, A new approach to fuzzy-rough nearest neighbour classification, in: Proceedings of the 6th International Conference on Rough Sets and Current Trends in Computing, Vol. 5, 2008, pp. 310–319.

[38] R. Jensen, C. Cornelis, Fuzzy-rough nearest neighbour classification and prediction, Theoretical Computer Science 412 (2011) 5871–5884.

[39] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The weka data mining software: an update, ACM SIGKDD explorations newsletter 11 (1) (2009) 10–18.

[40] I. Triguero, S. García, F. Herrera, Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study, Knowledge and Information Systems 42 (2) (2015) 245–284.

[41] N. Kasabov, Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning, IEEE Transactions on Systems, Man, and Cybernetics - Part B 31 (6) (2001) 902–918.

[42] N. K. Kasabov, Q. Song, Denfis: dynamic evolving neural-fuzzy inference system and its application for time-series prediction, IEEE Transactions on Fuzzy Systems 10 (2) (2002) 144–154.

[43] S. Soltic, N. K. Kasabov, Knowledge extraction from evolving spiking neural networks with rank order population coding, International Journal of Neural Systems 20 (6) (2002) 437–445.